EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
ORGANISATION EUROPEENNE POUR LA RECHERCHE NUCLEAIRE

CERN - PS DIVISION

PS/ PA/ Note 93-30

MAG-POL-FIT (MAGNET POLYNOMIAL FIT)

M. Chassard, R. Coccoli, L. Danloy, J. Delaprison, L. Durieu, G. Granger,
J.-Y. Hémery

Geneva, Switzerland
20 December, 1993

MAG_POL_FIT

Magnet Polynomial Fit

PS-PA note 93-30

M.Chassard, R.Coccoli, L.Danloy, J.Delaprison, L.Durieu, G.Granger, J-Y. Hemery

Table of contents

Introduction

At the date of editing this note, there are 70 different type of magnets in operation in the SOUTH and EAST experimental areas, as well as in several transfer lines of the PS complex. Four functions characterize each magnet when available. Those functions are the induction, its integral over length, the equivalent length as a function of the fed current, and the homogeneity as a function of coordinates in the magnet. Up to now, the data resulting from magnetic measurements were only available out of notes. A polynomial curve fitting has been applied on each of the 192 functions and is now available on computers. Software programs have been developed to first handle the huge amount of information, to fit the polynomial functions and to produce the plots, next to interrogate the data base for computations. The computations are $y=f(x)$ and $x=f^{-1}(y)$.

1. Data collection and processing

The data come from notes relative to magnetic measurements and stand as references in the input data files. The input data file also contains whether the data come from table or graph and the name of the operators who collected the data and performed the curve fitting.

There are 3 types of polynomial fit :

- Odd functions for the magnetic induction or gradient and their associated integrals (over z axis) versus current.
- Even functions for the equivalent magnetic length versus current, and homogeneity versus x or y axis.
- N^{th} order polynomial for all functions that are not symmetrical (ex: homogeneity of 'C' shape dipoles).

The order of the polynomial is chosen by the operator to obtain a relative error of less than 1% between experimental data and fitted function. The resulting function should be physically valid.

Upper limits are declared as constants in the software programs :

- number of input coordinates = 15
- order of polynomial = 13

n.b. before raising up the polynomial order close to the number of experimental points, one should check whether one or more points are obviously wrong. Neglecting typing errors, those bad points may come from measurements made on different hysteresis cycles. Furthermore, they prevent the function to be odd by not crossing at (0,0).

2. ADD a new type of magnet

Go through the sequence below to add one more magnet in the data base and to generate a new ALL.POLO file. Run the programs marked in "italic and underlined" character style on VM-CMS.

GIME *XZONE* (the XZONE disk should stand as your Z disk)

2.1. Create a new POLI file

XEDIT *fname POLI*
There is one example for dipoles and one for quadrupoles supplied in this document.
'Cname' is a string composed of 4 sub strings, and taking dipole 'D30A1' as an example:

D	dipole
30	30 cm long
A	a given type of 30cm long dipoles
1	Discrimination among D30A different types (gap,...)

'Fname' = file name on VM disk (limited to 6 characters)
'Cname' = computer name used in POLINV
'Mname' = names in the 'PROGRAMME HALL-SUD/EST' published by J. Delaprison (PS-PA-IN)

2.2. Update MFP-ALL.kumac

XEDIT *MPE-ALL KUMAC*

2.3. Find the place where to insert the new magnet, since they have been originally sorted by strength value (strongest first).

2.4. Fill the 'Cname', 'Mname' and the gap value on a line starting with the '*' character.

2.5. Enter one 'EXEC MFS 'Fname' line per record (record: as defined in POLI files).

2.6. Process data of all the POLI files, under PAW

EXEC MFP-ALL

2.7. Concatenate all the POLO files

COPYF * POLO F ALL POLO F (APPEND

2.8. Check consistency between ALL.POLO and MFP-ALL.KUMAC files and build MFP-ALL.LIST

MFPLIST

In case of mismatch, the program reports the names onto which it occurs.

2.9. Compute inverse polynomial function values (press PF10 in front of a POLY file)

RUNPOLY *a.POLY*
'a' is the name of the POLY file

3. INPUT and OUTPUT data files, and associated Programs

input_files		program		output_file
fname.**POLI**	>>>	mfp-all.kumac mfs.kumac magfit.ftn magnor.ftn magron.ftn magplt.ftn	>>>	fname.**POLO-*r***
fname.**POLO-*r***	>>>	copyf.exec	>>>	**ALL.POLO**
ALL.POLO	>>>			
	>>>	mfplist.exec	>>>	**MFP-ALL.LIST**
MFP-ALL.KUMAC	>>>			
ALL.POLO	>>>	runpoly.exec		
	>>>	polinv.exec	>>>	a.**POLX**
a.**POLY**	>>>	polinv.module		

'fname' = file name on VM
'r' = record number

4. POLI file format

There is one file per type of magnet and is to be in fixed format (refer to FFF.EXEC on PSRING).

A 'POLI file' may contains various record types (the units depend on the number of poles)

0 : force (T.m for dipoles or T for quadrupoles) as a function of current (A)
1 : equivalent magnetic length (m) as a function of force(T.m or T)
2 : field (T or $T.m^{-1}$) as a function of current (A)
3 : field homogeneity (% of max strength) as a function of radial position for a given current
4 : equivalent magnetic length (m) as a function of current (A)

4.1. Structure of a record

line 1

is the tag character to mark the first line of a record
record type number (refer to the above list)
record type string
data supplied by
data fitted by
type of source (Table or Curve)

#	0	BL	JD	JYH$	T

line 2

computer name of magnet
additional information
polynomial function string

D30A1	?(<gap >100<mm>)!	BL "M#T.<M>"N# = <F>(I "M#A"N#)$

line 3

number of coordinates
number of points for the plot on the horizontal axis
min. horizontal boundary for the plot
max. horizontal boundary for the plot
max. value for the fit
max. polynomial order
polynomial function type
4.1.1. : any
4.1.2. : odd function
4.1.3. : even function

6	300	.0	150.0	135.0	7.000	1.000

line 4 to line (4+np-1)

abscissa
ordinate
fitting weight (smaller value = higher weight for the fit)
comments

25.000	0.0190	.0001	blabla....

last line of the record

identification of the reference of the source from which the data have been collected:
ref AT/MA note 92/42

4.2. POLI file example for dipoles

Name of file on VM-CMS (Fname) : MN300A.POLI

All lines before the '#' sign character in column 1 are considered as comment lines.

```
#     0     BL     ..     JYH$     T
D30A1  ?(<gap >100<mm>)!          BL "M#T.<M>"N# = <F>(I "M#A"N#)$
      6    300    .000  150.000  135.000   7.000   1.000
    25.000       0.0190          .0001
    50.000       0.0380          .0001
    75.000       0.0568          .0001
   100.000       0.0750          .0002
   125.000       0.0922          .0003
   150.000       0.1047          .0005
ref AT/MA note 92/42
#     1     L     ..     JYH$     T
D30A1  ?(<gap >100<mm>)!          L "M#<M>"N# = <F>(BL "M#T.<M>"N#)$
      7    525    .000   0.105    .105   10.000   2.000
     0.0050       0.3830          .0001   *
     0.0190       0.3830          .0001
     0.0380       0.3830          .0001
     0.0568       0.3830          .0001
     0.0750       0.3825          .0003   *
     0.0922       0.3815          .0003   *
     0.1047       0.3800          .0001
     0.0750       0.3820          .0002
     0.0922       0.3820          .0002
#     2     B     ..     JYH$     T
D30A1  ?(<gap >100<mm>)!          B "M#T"N# = <F>(I "M#A"N#)$
      6    300    .000  150.000  135.000   7.000   1.000
    25.000       0.0495          .0001
    50.000       0.0990          .0001
    75.000       0.1482          .0001
   100.000       0.1964          .0002
   125.000       0.2414          .0002
   150.000       0.2757          .0003
#     3     H     jyh     JYH$     C
D30A1  ?(<gap >100<mm>, 150A)!          [E]> ("Y#) = <F>(X "M#<MM>"N#)$
      10    500    .000  100.000  100.000   8.000   2.000
     0.002       -0.0010          .0001
    10.000       -0.0100          .0001
    20.000       -0.1000          .0001
    40.000       -0.6300          .0001
    50.000       -1.2600          .0001
    60.000       -2.3160          .0001
    70.000       -3.5860          .0002
    80.000       -6.9500          .0002
    90.000       -11.3700         .0003
   100.000       -18.0000         .0003
```

4.3. POLI file example for quadrupoles

Name of file on VM-CMS (Fname) : **QD.POLI**

```
#     0     GL     ..     JYH$      T
Q82                              GL "M#T"N# = <F>(I "M#A"N#)$
     10     555    .000  550.000  550.000    5.000    1.000
    100.7600     3.6040           .0001
    148.8400     5.3110           .0001
    200.5400     7.1450           .0001
    250.5100     8.9130           .0001
    300.4000     10.6700          .0001
    350.2400     12.3800          .0001
    400.7100     13.9800          .0002
    450.8000     15.2400          .0002
    501.4800     16.2900          .0003
    553.2900     17.1900          .0003
Ref.ISR/MA     QD
#     1      L      ..     JYH$      T
Q82                           L "M#<M>"N# = <F>(G "M#T.<M^>-1!"N#)$
      7     202    .000   20.000   20.000    4.000    2.000
     4.1703      0.8642           .0001
     8.2696      0.8640           .0001
    10.3303      0.8628           .0001
    14.3886      0.8604           .0001
    16.3222      0.8565           .0001
    19.1040      0.8527           .0002
    20.1807      0.8518           .0003
#     2      G      ..     JYH$      T
Q82                           G "M#T.<M>^-1!"N# = <F>(I "M#A"N#)$
      7     550    .000  550.000  550.000    5.000    1.000
    100.7600     4.1703           .0001
    200.5400     8.2696           .0001
    250.5100     10.3303          .0001
    350.2400     14.3886          .0001
    400.7100     16.3222          .0001
    501.4800     19.1040          .0001
    553.2900     20.1807          .0002
#     3      H      ..     JYH$      .
Q82                             [E]> ("Y#) = <F>(X "M#<MM>"N#)$
      p    nbins    .000   10.000    0.000    7.000    2.000
     0.0000      0.0000           1.0000
#     4      L      ..     ...$      .
Q82                              L "M#<M>"N# = <F>(I "M#A"N#)$
      0     900    .000  100.000  100.000    8.000    2.000
    100.0000     1.23456          0.0001
```

5. POLO file format

File structure (example below given for 6 points data in '.POLI' file and polynomial order 7)

line 1 : copy of line 1 of POLI file
line 2 : copy of line 2 of POLI file
line 3 : max x amplitude...polynomial type(any=0, odd=1, even=2)...equivalent length
line 4 : number of coefficients of the polynomial (8 for this example)
line 5 : coefficient of x^0
.
.
line 12 : coefficient of x^7

line 13-18 : input data list ... error in % between input data and the fit
line 19: rms relative error between the 6 points and the fit

Example

Name of file on VM-CMS (Fname) : **D30A1-0.POLO**

```
#     0     BL     ..     JYH$     T
D30A1  ?(<gap >100<mm>)!          BL "M#T.<M>"N# = <F>(I "M#A"N#)
 0.15000E+03 0.10000E+01 0.10475E+00
       5.1.
 0.00000E+00
 0.76215E-03
 0.00000E+00
-0.10731E-08
 0.00000E+00
 0.60731E-13
 0.00000E+00
-0.61817E-17

  5.2.   25.0000        0.0190        0.0190        0.1973%
  5.3.   50.0000        0.0380        0.0380       -0.0328%
  5.4.   75.0000        0.0568        0.0568       -0.0525%
  5.5.  100.0000        0.0750        0.0751        0.1744%
  5.6.  125.0000        0.0922        0.0921       -0.1319%
  5.7.  150.0000        0.1047        0.1048        0.0482%
 Rms relative error :  0.1244%
```

6. POLY file format

This is the data used by POLINV and allows to compute the current and force relationship from the polynomial fits stored in the file ALL.POLO. It consists of an input deck specifying the problem at hand.

- Any line of which the first non-blank character is not alphanumeric ['0'..'9','A'..'Z','a'..'z'] will be considered as a comment. The same holds for any empty line. Items are separated by whitespace, that is blank or tab character(s). Any more data existing on a line once the relevant items have been read is taken as comment.
- The deck is terminated by a line containing EOR in column 1 (or by reaching the end-of-file mark).
- The active part of the problem consists, in sequence :
  - Title line whose content is arbitrary, however it should exist once and only once.
  - Operation and loop control : specify the way to solve the problem, contained on one line :
    - computation mode (now 0, 1 or 2)
    - start value of the loop (described later for each mode)
    - step value (increment to add at start value at each step, same units and significance)
    - number of steps (0 means no execution at all!)
    - normalization value for the given forces (see each mode).
  - Followed by a set of records, specifying the magnets on which to run. The number of those records is at present limited to 20 due to memory allocation in the program. They look like :

    L.name M.name C.name ref_value

    Where the various elements should exist :
    - L.name : Alphanumeric name of the magnet, for example E17SBHZ03 (will be displayed on output)
    - M.name : The computer name of the same magnet, for example MEP23.04
    - C.name : The name of the reference magnet as described in the data base, D250D
    - ref_value: A real number, used as starting value for computation.

6.1. Computation mode 0

Computes currents from force requirement (T for quadrupoles, T.m for dipoles, A for intensity).

- start value of the loop : first value of the momentum for which the current is required
- normalization value : momentum used to define the ref_values in the magnet records
- ref_value : magnet strength required at the 'normalization' momentum

For example, running a beam optics program at a reference momentum of 10.0 GeV/c, we want to know the required currents at 1.0 and 1.5 GeV/c. The input deck will look like this :

```
Title : Currents at 1.0 and 1.5 GeV/c from forces computed at 10 GeV/c by TRANSPORT
! This line is a comment and will be ignored by POLINV. The title line is mandatory!
! This line is also a comment, as are the next two

+mode  start   increment      steps   normalization
  0     1.0       0.5           2          10.0
ZT7QFO03      Q123   Q120   1.8245      unit is T, from TRANSPORT run at 10 GeV/c
ZT7QFO05      Q125   Q120   2.1234      idem
T7QFO01       QFS2   Q80    4.17        idem
EOR
```

6.2. Computation mode 1

Computes force from magnet current; then multiplies by 'normalization'.

This is the force to magnet current relationship when 'normalization' is set to 1. Proper use of 'normalization' allows to compute deflection or focus length in any convenient unit.

Here are two examples for a bending magnet:
'normalization' set to 1 ⇒ result in T.m
'normalization' set to 1/Bρ ⇒ result in radian

6.3. Computation mode 2

Identical to mode 0, except for the 'ref_value' that is a value of current in Amps instead of a force. This allows momentum scaling from an experimental setting.

Special output cases *(valid for all computation modes)*

- '????' : Means that the force polynomial could not be found. This usually means a C.name misspelling (they are case sensitive).
- '****' : The computed value is out of range. The magnet can not provide the required force.
- 'value*' : The search program did not converge and the value cannot be trusted.

7. POLX file format

On execution the RUNPOLY exec file will produce a POLX file. The units in the output file depend on the calculation mode requested in the POLY file.

Example : PS201X0.POLX

Momentum	e6d4	e6f5	x0d1	x2f2	x2n3	m2f3	m2d4
1.0	-11.86	69.29	49.40	-5.96	10.65	13.20	-14.79

The current in ampere are :

e6d4	:=	-11.86
e6f5	:=	69.29
x0d1	:=	49.40
x2f2	:=	-5.96
x2n3	:=	10.65
m2f3	:=	13.20
m2d4	:=	-14.79

8. MFP-ALL.LIST file format

The file is added as appendix A. Its content being valid at the time of publication. The up-to-date version is available on PSRING disk under VM-CMS and named MFP-ALL LIST.

This file consists in 3 blocks : header, data and trailer.

① *The **header** supplies the date and time of creation and the mnemonics for each column of the data table.*

MFP list: 20 Oct 1993 at 15:51:16 (PS-PA-EA)

|C.name	**|M.name**	**|F.name**	**|rec**	**Imax**	**Strgth**	**Gap**
|.....	|.....	|.....	|..	|..	|..	|..

Meaning of the mnemonics:

'Cname' = computer name used in POLINV
'Mname' = names in the 'PROGRAMME HALL-SUD/EST' published by J. Delaprison (PS-PA-IN)
'Fname' = file name on VM-CMS disk (limited to 6 characters)
'rec' = list of record types which are available for this given magnet
'Imax' = maximum current in Amps for which the magnetic measurements have been performed.
'Strgth' = strength in T or T.m at the current mentioned above
'Gap' = size of the gap between the poles in mm

② *The **data** is consistent with the above field description and is one line per magnet.*

D50B1	**MNPA50b1**	**MN50B1**	**0 1 2 3**	**600**	**0.5017**	**125**

'Cname' = D50B1
'Mname' = MNPA50
'Fname' = MN50B1
'rec' = records 0...3 are available from the MNP50B1.POLI file
'Imax' = 600A is the maximum current to be set in normal operation
'Strgth' = the strength is 0.5017 T.m at 600A
'Gap' = the gap between the 2 poles is of 125mm

③ *The **trailer** gives the audit of the amount of available data in MFP-ALL.POLO*

70 different types of magnets have been registered (192 records)

9. POLINV program description

POLINV is a solving program that is able to find a solution to an equation in the form : $y = f(x)$. Where $f(x)$ is a polynomial function in x. The program will solve for y when x is given (trivial) or for x when y is given provided some mathematical constraints are met. It has been written in C in a portable manner and has been compiled and run on several machines with no problems : VM-CMS, IBM-PC under DOS, SUN IPC, VAX under VMS and AMIGA.

It is built out of two main modules :

① Reads the information contained in ALL.POLO and stores it in memory in a computer usable way, making consistency checks and some internal evaluation (definition boundaries) in the way.

② Reads the elements of the problem and solves as required when possible. Reports the results, tagging appropriately dubious or non-solvable items.

The two modules are merged in a single program under VM-CMS, directly using the file ALL.POLO in its text form. They are split in two programs : MAKEBIN.C and POLINV.C on the other machines, mostly due to segmentation problems on the IBM-PC, it also makes execution faster.
MAKEBIN will take the ALL.POLO text file and generate a machine specific binary equivalent file named MAGNETS.BIN for later use by POLINV. The intermediate file contains all the required derived information, is much smaller and loads quicker.

Warning : The ALL.POLO and the .C files are straight text and can be transported between different architectures (with FTP for example). The executables and the MAGNETS.BIN files are machine specific and have to be generated on the target. Any change in the ALL.POLO file imposes the regeneration of the MAGNETS.BIN file (except on VM-CMS).

Algorithm :

- A function can only be inverted in an interval where it is single-valued. To ensure this, the loader part of the program will examine the given polynomial (records #0 and #2 only) to make sure the derivative has no zero in the specified interval. If not, it will compute the location of this extremum and replace the maximum abscissa by the one of the extremum. The program then computes the ordinate of this point. The check is quite crude and not general at all. The program only works on the force record (record #0 from the .POLI file). This record is always an odd function and a simple check is enough.
- The zero derivative is checked by looking at the derivative at zero and at the interval maximum by ensuring the signs are the same. This only guarantees that the number of real zeros of the derivative in the interval is even. Although not complete, it is sufficient for our intended purpose. It should however be borne in mind, should the program need extension at a later time.
- Solution of $y = f(x)$, y given, x unknown : Uses an Euler algorithm; an initial value of x is guessed, then the value of the function and its derivative are computed for this guess. Next a better guess of x is computed from the function value, the required value and the local derivative. After several tries, the x value should converge to the required solution.
- Exception handling :
  - Magnets are known by name and the given name must obviously exist in the database. The record #0 should also exist and be valid. Otherwise the program exits and reports the error.
  - The program first checks that the required force is within bounds, if not it exits at once, if yes it makes a first linear guess and proceeds with the convergence routine.
  - In order to avoid infinite loops, the recurrence computation of x is tried a maximum of 9 times or until the relative error is less than 10ppm. If these conditions fail, the program reports the last try and the lack of convergence.

Built-in limitations and sources of trouble :

The maximum number of magnet records that can be handled in one problem is *limited to 20* due to memory management. The output format becomes awkward well before that; a good idea is to limit oneself to around 8 magnets.

Due to temporary storage definition, the input lines should not be longer than *98 characters* including trailing spaces. Failure to comply can give some nasty results.

Equations solved mode by mode : identities are solved for result

For the mathematically minded person, POLINV tries to satisfy numerically one of the following equations by altering the value of 'result'.

mode 0 : (current_value/normalization)*ref_value = f(result)
mode 1 : result = f(current_value*ref_value)*normalization
mode 2 : (current_value/normalization)*f(ref_value) = f(result)

Where the symbols used have been defined in the input deck description and current_value is :
start + increment*n (n varying from 0 to step-1 inclusive).

Where to find the various files :

On VM-CMS :

ALL.POLO	user XZONE (repository data set), copy on user PSRING
POLINV.MODULE	user PSRING, full copy on user XZONE
call template	POLINV inputfile[filetype[filemode]] [outputfile[filetype[filemode]]
or easier	type 'POLINV' in front of the source file (under FILELIST or equivalent)

The default input filetype is POLY, the default output filetype is POLX. If no output file is specified, it will have the same name as the input with filetype of POLX.

On the PS network :

ALL.POLO	not kept, unused	
MAGNETS.BIN	g:\home\d\durieu\magnets	needed
MAKEBIN.C	g:\home\d\durieu\c_source	
POLINV.C	g:\home\d\durieu\c_source	
MAKEBIN.EXE	g:\home\d\durieu\magnets	
POLINV.EXE	g:\home\d\durieu\magnets	needed

On other machines :

Not supported by the PS/PA/EA.
If you are interested, you should import the source files from the PS network and compile them on your target. You should also import the magnet database from VM_CMS and run MAKEBIN. You are then on your own for keeping synchronous with potential updates.

Appendix A

MFP list: 20 Dec 1993 at 11:57:52 (PS-PA-EA)

C.name	M.name	F.name	rn				Imax	Strgth	Gap
"D250	MHB1	MHB1	0	1	2		1450	3.3992	80
"D250A	MHB2	MHB2	0				400	3.1619	80
"D250B	MVB1	MVB1	0	1	2		1550	2.9638	108
"D250C	MVB2	MVB2	0				400	2.4930	108
'D250D	MNP23/MEP23	MNP23	0				1500	4.0217	60
"D220	MVB4	MVB4	0	1	2		400	2.2121	108
'D200A	M200TP	M200TP	0	2			800	3.7564	140
'D200B	M200SP	M200SP	0	1	2		800	3.5862	140
'D200C	MC200	MC200	0	1	2		850	3.4789	140
'D200D	MNP35 SPL.E	MNP35	0				360	1.9890	20
"D185	MHB3	MHB3	0				400	2.3743	80
'D150	MEJ150	MEJ150	0	1	2		880	3.5193	50
"D150A	MCS01	MS01	0	1	2	3	1500	2.0900	140
"D140	MHB5	MHB5	0	1	2		400	1.8368	80
'D115	SPL.SUD	SPLIT	0	2			700	0.8613	70
'D105	M105	M105	0	1	2	3	600	0.2598	100
'D100	MEJ100	MEJ100	0	1	2		900	2.1550	40
'D100A	M100TP	M100TP	0	2			800	2.0868	140
'D100B	M100SP	M100SP	0	1	2		800	1.9718	140
'D100C	MC100	MC100	0	1	2		850	1.8283	140
'D100D	MSEP (sold)	MSEP	0	2			310	0.0717	???
"D100E	MVB3	MVB3	0	1	2		400	1.0789	108
"D100F	MHB4	MHB4	0	1	2		400	1.3432	80
"D62A1	MC6201	MC6201	0	1	2		210	0.2868	100
"D62A2	MC6202	MC6202	0	1	2		200	0.3010	100
"D62A3	MC6203	MC6203	0	1	2		200	0.2872	100
"D62A4	MC6204	MC6204	0	1	2		200	0.3073	100
'D50A	MNP39	MNP39	0	1	2		150	0.1757	100
"D50B1	MNPA50b1	MN50B1	0	1	2	3	600	0.5017	125
'D50B2	MNPA50b2	MN50B2	0	1	2		600	0.3248	200
'D45	M45	M45	0	1	2		300	0.3700	100
"D43	MEA43	MEA43	0	1	2	3	220	0.2215	220
'D40	MDX52	MDX52	0	2			250	0.7757	52
'D40A	MDX80	MDX80	0	2			250	0.6077	80
'D40B	MDX100	MDX100	0	2			260	0.5392	100
'D38	MNPA38	MNP38	0	1	2	3	280	0.2206	200
"D30A1	MNP300	MN300A	0	1	2	3	150	0.1048	100
"D30A2	MNP300	MN300B	0	1	2	3	150	0.0598	200
"D30B	Dip.B.30	D30	0	1	2	3	16	0.0304	150
"D30C	Dip.Lear	DLEAR	0	1	2	3	12	0.0209	220
"D30D	Dip.B.28.h	D28H	0	1	2		10	0.0081	170
"D30E	Dip.B.28.v	D28V	0	1	2		10	0.0079	140
'D30F	MNPA30	MNPA30	0	2			600	0.2443	205
'D30G	M30	M30	0	2			600	0.2155	200
'D25	MNPA25	MNPA25	0	2			650	0.2107	202
'D19A	MNPA19	MNPA19	0	2			250	0.0901	200
'D19B1	MEA19	MEA19	0				180	0.1494	130
'D19B2	MEA19A	MEA19A	0	2	4		180	0.1179	196
'Q200	Q200	Q200	0	1	2		800	22.9800	200
'Q150	Q150	Q150	0	2			900	15.5090	200
'Q120	Q12	Q12	0	1	2		800	61.3230	50
'Q120A	Q120	Q120	0	1	2	4	900	31.2510	100
'Q120B	QFL	QFL	0				560	24.5120	100
'Q100	Q100	Q100	0	1	2		700	12.1580	200
"Q82	QD	QD	0	1	2		550	17.0870	91
'Q80	QFS	QFS	0	1	2		560	16.5070	100
'Q75	Q75	Q75B	0	1	2		850	39.3250	50
'Q75A	Q800	Q75A	0	1	2		820	9.0084	200
"Q74	Q74	Q74	0	1	2	3	830	37.4970	50
'Q50	Q50	Q50	0	1	2		650	6.6953	80
"Q50A	Q50-Isolde	Q50ISL	0	1	2	3	300	5.5723	130
"Q50B	Q500	Q500	0	1	2		620	6.6463	200
'Q42	Q42	Q42	0	1	2		11	1.1240	130
'Q40	QLC/QNP02	QLC	0	2	4		1000	3.1313	250
"Q40A	QLG	QLG	0	1	2		800	3.1795	250
'Q30	QTN	QTN	0				200	2.7122	184
'Q30A	QTS	QTS	0				200	2.4853	184
"Q30B	QLD	QLD	0	1	2	3	800	3.0201	250
'Q25	Q25	Q25	0	2			580	3.4901	200
'Q22	Q22	Q22	0	1	2		400	3.0307	200

70 different types of magnets have been registered (192 records)